EMI

المدرسة المحمدية للمهندسين

Ecole Mohammadia d'Ingénieurs

Smart optimization for mega construction projects using artificial intelligence

Presented by: GHANIMI Oussama

Supervised by professor: Ms EL MKHALET Mouna

Plan

- **Introduction:**
  - **why megaprojects fail?**
  - **Purposes.**
- **Employed techniques.**
- **Decision’s variables.**
- **Model implementations (incorporation of the basic concepts of Critical Path Method ‘‘CPM’’ with a multi-objective Genetic Algorithm ‘‘GA’’ simultaneously):**
  - **Initialization.**
  - **Evaluation.**
  - **Generation.**
- **Smart Critical Path Method System “SCPMS”.**
- **Case study and the added value.**
- **Possibility of applying this optimization model in Morocco.**
- **Conclusion.**
- **References.**

➢Introduction:

Mega infrastructure projects are large-scale engineering facilities providing fundamental public services for social production, economic development, and people's life, such as large-scale hydropower projects, high-speed railways, expressway networks, gas pipeline projects, and long-span bridges. Many mega infrastructure projects have been or are being built around the world, such as Hong Kong–Zhuhai–Macao bridge, The Su-Tong Yangtz River bridge, high-speed railway in China, and Darwin Impex LNG plant.

Traditional project management is based on three-dimensional life cycle approach where the project manager is to seek to optimize the cost, quality, and design. Different from traditional construction project management, construction of mega infrastructure projects exhibits high complexity, such as task complexity, structural and directional complexity, technical complexity, and organization complexity. Due to their complexities, many of mega infrastructure projects suffer from overbudget, overtime, and low investment return. in the same vein according to the study, published in *Project Management Journal,* a team of academics list the six themes as decision-making behaviour; strategy, governance and procurement; risk and uncertainty; leadership and capable teams; stakeholder engagement and management; supply chain integration and coordination. They were found to be all of equal importance when analysing why such projects seem doomed to fail.

Talking numbers, the Oxford Global Projects Database (OGPD) – which holds information on nearly 12,000 different megaprojects, stated conforming to their own data, nearly every category of megaprojects failed to meet planned budgets, and to be completed on time. This includes nearly 40%of rail projects and 61% of projects in the aerospace sector. Even more distressingly, only a tiny 0.5% of megaprojects provided all of the ‘extended benefits’ they were initially planned to deliver.

• Purposes.

As a response to the construction problems previously discussed, this optimization seeks to confirm the following objectives that are used for the mega construction projects: (1) development of a model which incorporates the basic concepts of Critical Path Method ‘‘CPM’’ with a multi-objective Genetic Algorithm ‘‘GA’’ simultaneously; (2) development of proposed software which is named Smart Critical Path Method System,‘‘SCPMS’’ using features of genetic algorithms; two things that will be helpful in planning, scheduling and controlling the mega construction projects that hold a strong promise to:(a) increase resource use efficiency; (b) reduce construction time;(c) minimize construction total cost (direct cost plus indirect cost); and (d) measure and improve the quality of future mega construction projects

➢Employed techniques

The model formulated proposed the following technique: (1)Activity On Node ‘‘AON’’ is used to simulate the main project as shown in Fig. 2; (2) Activity On Node ‘‘AON’’ is also used to simulate the sub-projects as shown in Fig. 3; (3) the activity (k),(where k=1,2, . . . ,K) that in case of repetitive networks, LOB is used to simulate the repetitive activity schedule into sub-pro-jects; (4) each activity (k), has a time buffer (TBk,kk), at each stage(s), (where s=1,2, . . . ,S) between completion time of first activity (k) with start time of each following activity (kk)in the sub-project network as shown in Fig. 4; (5) any two sequential activities at sub-project may have a stage buffer(SBk,kk), of a specific number of stages at any unit time to meet practical and/or techno running purposes, this stage’s buffers have to be identified by the planner for these activities; (6) a discrete relationship between duration, cost, and quality of each activity (k) in the network of sub-projects; (7) transformation of traditional LOB to modified CPM must be done in proposed model; and (8) genetic algorithm tool is used to achieve the optimum results .

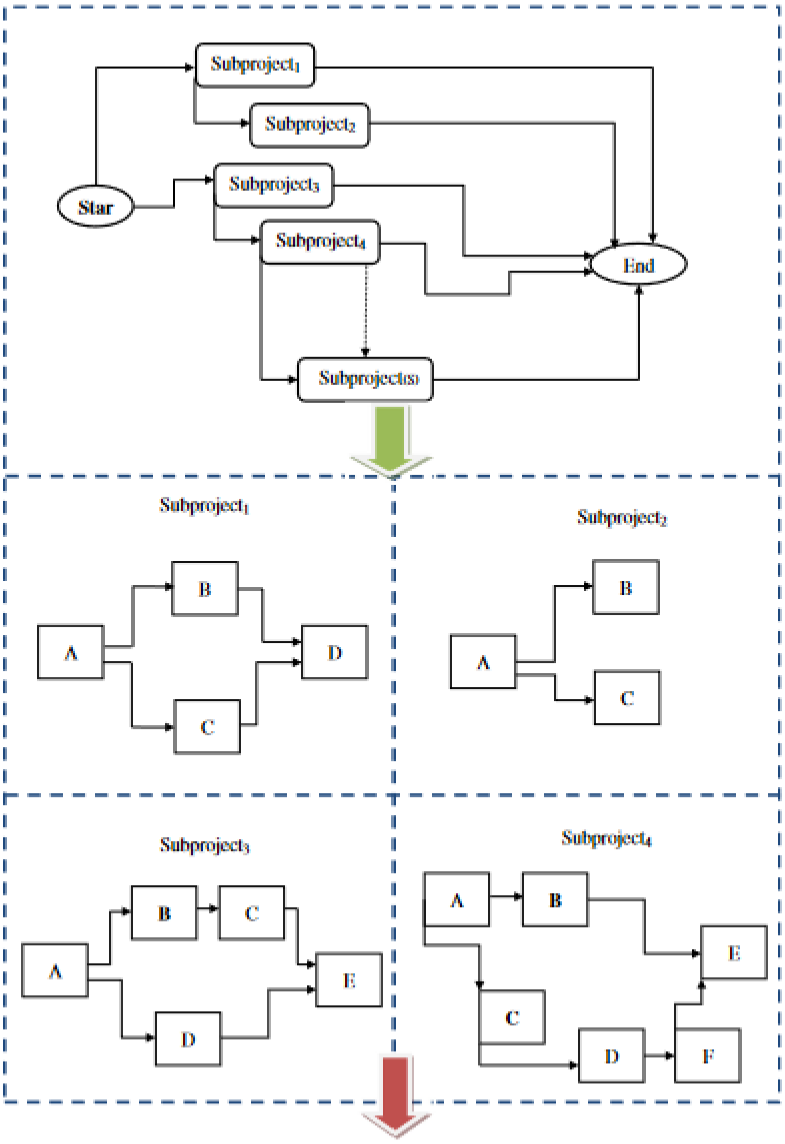


Figure 2

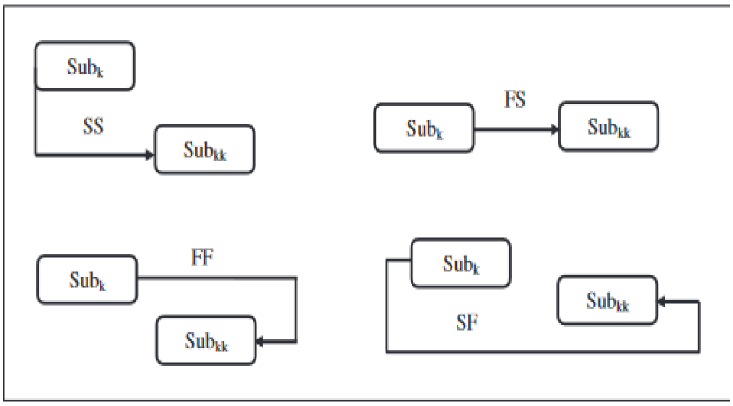


Figure 3 Available lag values between two sequential sub-projects.

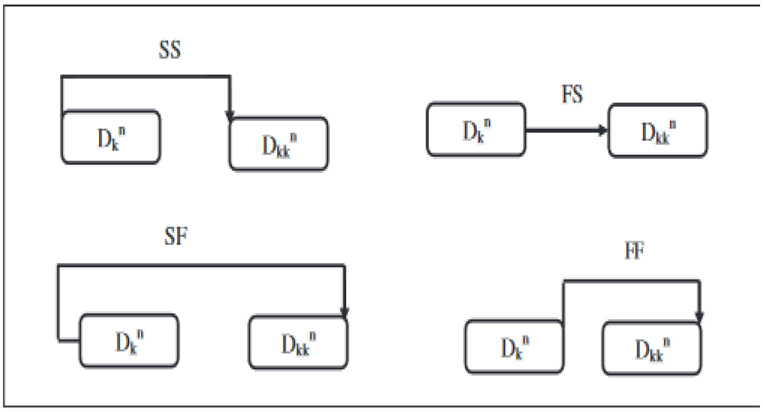


Figure 4 Available lag values between two sequential activities.

➢Decision’s variables

All relevant decision variables are represented In Fig 5. Decision makers can create more alternatives, with named resource utilization for each activity and change from alternative to another considering permutations of alternatives by selecting one random option of each previous five main decision variables and grouped together in a single alternative, first creation of resource utilization takes n= 1, and so on with repeating the random selections considering permutations of options of each previous five main decision variables till reaching last creation of resource utilization takes n=N. Where is the number of available resource utilization options for each activity into the studied project as shown in Fig 5. The enumeration of all possible combinations of these available resource utilization options at the activity level can lead to a total of more project alternatives, which means largescale of feasible construction resource utilization plans at the project level.

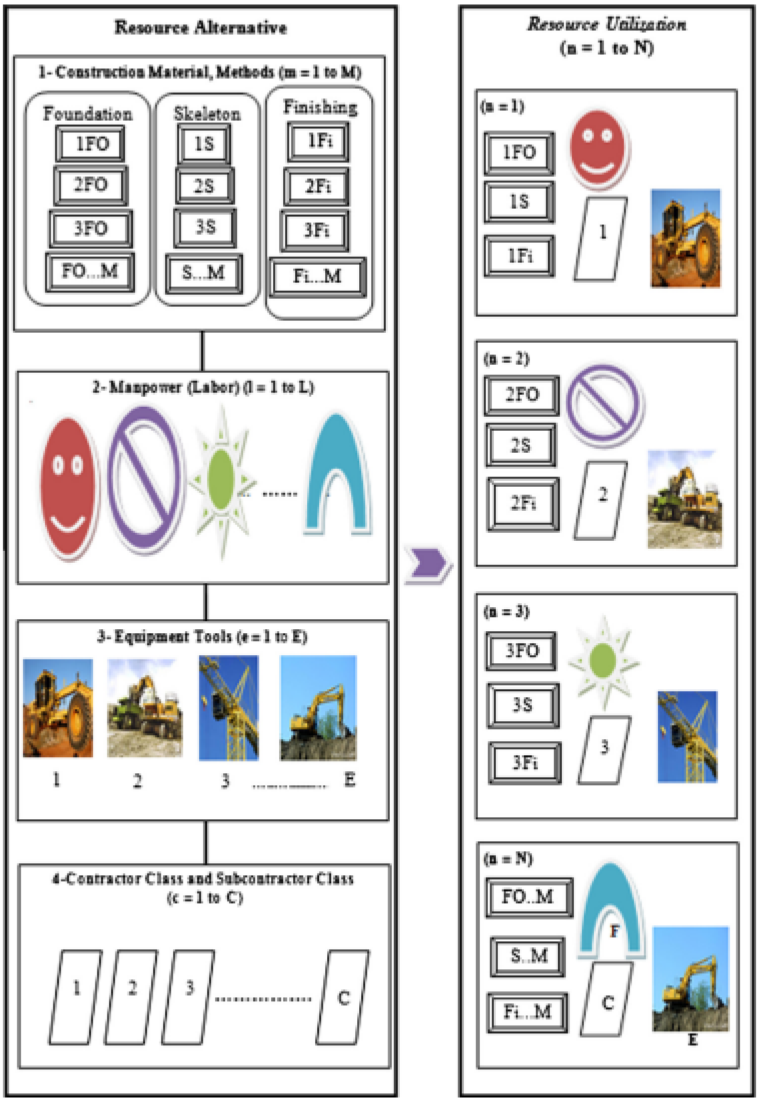

Resource Alternative
1- Construction Material, Methods (m = 1 to M)
Foundation
1FO
2FO
3FO
FO...M
Skeleton
1S
2S
3S
S...M
Finishing
1Fi
2Fi
3Fi
Fi...M
2- Manpower (Labor) (l = 1 to L)
......
3- Equipment Tools (e = 1 to E)
1
2
3
E
4-Contractor Class and Subcontractor Class (c = 1 to C)
1
2
3
C
Resource Utilization (n = 1 to N)
(n = 1)
1FO
1S
1Fi
1
(n = 2)
2FO
2S
2Fi
2
(n = 3)
3FO
3S
3Fi
3
(n = N)
FO..M
S..M
Fi..M
F
C
E

Figure 5 Utilization of resources alternative options.

➢Model implementations

The proposed model was implemented in three major stages: (1) initialization that generates an initial set of (u) possible solutions for this resource utilization problem; (2) evaluation that calculates the time, cost, and quality of each generated solution; and (3) generation that seeks to improve the fitness of solutions over successive generations

▪ Initialization

Number of generations and population size are identified based on selected string size in order to decrease the time and cost and to improve the quality of the solution. These solutions represent an initial set of resource utilization options that can be used to construct each and every activity in mega projects. This initial set of possible solutions is then evolved in the following two stages in order to generate a set of optimal resource utilization options for each activity in mega project that establishes optimal trade-offs between construction time, cost and quality.

▪ Evaluation

The main purpose of this stage is to evaluate the time, cost, and quality of each possible solution (u) in generation (g) in order to determine the fitness of the solution. This fitness determines the likelihood of survival and reproduction of each solution in the following generations. This stage evaluates the three identified fitness functions for each solution using the following steps: (1) Calculate project duration for solution (u) in generation (g), which consists of total duration of all activities on the critical path. In this model, project duration is estimated using newly developed algorithms for integrating the scheduling of repetitive activities LOB with non-repetitive activities CPM for mega construction projects as shown in Eqs. (1)– (10);

$$Z_1^n = \text{Min}\sum_{k=1}^{k=K}\{T_k^n + (FS_{k,kk}{}^n)_m\} \tag{1}$$

$$T_k^n = S \times D_k^n \tag{2}$$

$$D_k^n = Qt_k^n \div PR_k^n \tag{3}$$

$$(FS_{k,kk}{}^n)_m \geq TB_{k,kk}^n + (1 - S) \times D_k^n \tag{4}$$

$$(FS_{k,kk}{}^n)_m \geq TB_{k,kk}^n + (1 - S) \times D_{kk}^n \tag{5}$$

$$(FS_{k,kk}{}^n)_m \geq (SB_{k,kk} - S) \times D_k^n \tag{6}$$

$$(FS_{k,kk}{}^n)_m \geq (1 + SB_{k,kk} - S) \times D_{kk}^n - D_k^n \tag{7}$$

$$TB_{k,kk}^n = SS_{k,kk} - D_k^n \tag{8}$$

$$TB_{k,kk}^n = FF_{k,kk} - D_{kk}^n \tag{9}$$

$$TB_{k,kk}^n = SF_{k,kk} - (D_k{}^n + D_{kk}^n) \tag{10}$$

Where Z1n= Total Project Duration in (days) using resource utilization (n),Tkn= Total durations in (days) at all stages of an activity (k) on the critical path at new CPM using resource utilization (n), (FSk,kkn)m= Modified finish to start at a new CPM between two sequential activities using resource utilization (n),S= Number of stages in the project,Dkn= Duration in (days) at one stage of an activity (k) using resource utilization (n),Qtkn= Quantity of an activity (k) at one stage using resource utilization (n),PRkn= Production rate of an activity(k) at one stage using resource utilization (n),Dkkn= Duration in (days) at one stage of a successor activity (kk) at LOB using resource utilization (n),SBk,kk= Stage buffer between the starts of two sequential activities (k) and (kk) at LOB ,TBk,kkn= Time buffer in (days) between two sequential activities finish (k) and start (kk) at LOB using resource utilization(n),SSk,kk= Start to Start in (days) between two sequential activities (k) and (kk) at LOB ,FFk,kk= Finish to Finish in(days) between two sequential activities (k) and (kk) at LOB,SFk,kk= Start to Finish in (days) between two sequential activities (k) and (kk) at LOB; (2)Estimate project cost for solution (u) in generation (g), which consists of material, crew and subcontractor costs as shown in Eq.(11)

$$Z_2^n = \text{Min}\left\{ S \times \sum_{k=1}^{k=K} (M_k{}^n + D_k^n \times R^n + B_k^n) + (IC \times Z_1^n) \right\} \tag{11}$$

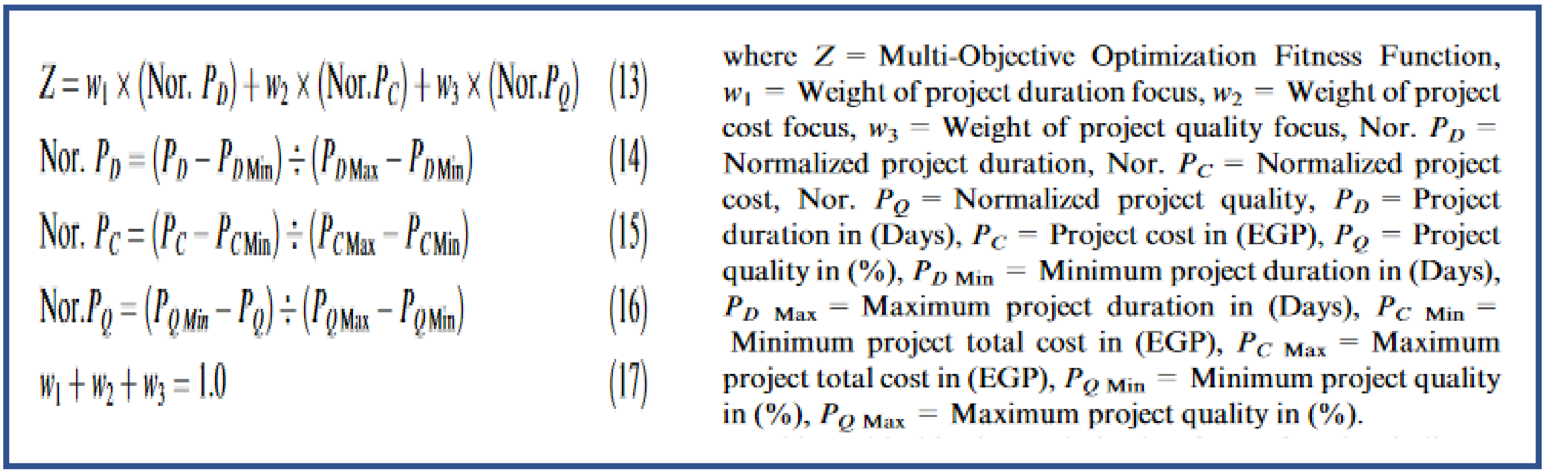
whereZ2n= Total Project {Direct} + {Indirect} Cost in(EGP) using resource utilization (n),S= Number of stages in the project,Mkn= Material cost in (EGP) at one stage of an activity (k) using resource utilization (n),Dkn= Duration in (days) at one stage of an activity (k) using resource utilization (n),Rn= Daily cost rate in (EGP/Day) of an activity (k)using resource utilization (n),Bkn= Subcontractor lump sum cost in (EGP) at one stage of an activity (k) using resource utilization (n) if found, IC = Daily indirect cost rate in (EGP/Day) throughout project life; and (3) Measure project quality (PQgu) for solution (u) in generation (g) as shown in Eq. (12).

$$Z_3^n = \text{Max}\left\{ \sum_{k=1}^{k=K} wt_k \times \left(\sum_{i=1}^{i=I} wt_{k,i} \times Q_{k,i}^n \right) \right\} \tag{12}$$

Where Z3 n = Total Project Quality in (%) using resource utilization (n), wtk = Weight of an activity (k) compared to other activities in the project, wtk,i = Weight of quality indicator (i) compared to other indicators in the activity (k), Qk,i n = Performance of quality indicator (i) in (%) at activity (k) using resource utilization (n).

- ## Generation

The purpose of this stage is to create three types of population in each of the considered generations: (a) parent; (b) child: and (c) combined. For each generation (g), a parent population (Pg) is used to generate a child population (Cg) in a similar manner to that used in traditional GAs, Goldberg [16]. The purpose of generating this child population is to introduce a new set of solutions by rearranging and randomly changing parts of the solutions of the parent population. This child population can then be combined with the parent population to create an expanded set of possible solutions that forms the combined population (Ng) for generation (g). This combined population (Ng) is used to facilitate the comparison between the initial solutions in the parent population and those generated in the child population. The best solutions in this combined population regardless of their origin are retained and passed to the following generation as a parent population. At this stage the procedure is implemented in the following steps:(1) Calculate optimal rank for each solution (u = 1–U) in the parent population (Pg). First, this is done by ranking solutions in the population according to their highest quality and lowest cost and duration, this is called Pareto optimal domination of solutions, where a solution is identified as dominant if it is better than all other solutions in all considered optimization objectives simultaneously. Second, this step calculates the multi-objective fitness function of each solution as shown in Eqs. (13)– (17), which represents equivalent normalized fitness of each corresponding project time-cost-quality solution. The fitness values help the algorithm to spread the obtained solutions over a wider optimal front and select best range of optimal solutions;



$$Z = w_1 \times (\text{Nor. } P_D) + w_2 \times (\text{Nor.} P_C) + w_3 \times (\text{Nor.} P_Q) \quad (13)$$

$$\text{Nor. } P_D = (P_D - P_{D\,\text{Min}}) \div (P_{D\,\text{Max}} - P_{D\,\text{Min}}) \quad (14)$$

$$\text{Nor. } P_C = (P_C - P_{C\,\text{Min}}) \div (P_{C\,\text{Max}} - P_{C\,\text{Min}}) \quad (15)$$

$$\text{Nor.} P_Q = (P_{Q\,\text{Min}} - P_Q) \div (P_{Q\,\text{Max}} - P_{Q\,\text{Min}}) \quad (16)$$

$$w_1 + w_2 + w_3 = 1.0 \quad (17)$$

where Z = Multi-Objective Optimization Fitness Function, w_1 = Weight of project duration focus, w_2 = Weight of project cost focus, w_3 = Weight of project quality focus, Nor. P_D = Normalized project duration, Nor. P_C = Normalized project cost, Nor. P_Q = Normalized project quality, P_D = Project duration in (Days), P_C = Project cost in (EGP), P_Q = Project quality in (%), $P_{D\,\text{Min}}$ = Minimum project duration in (Days), $P_{D\,\text{Max}}$ = Maximum project duration in (Days), $P_{C\,\text{Min}}$ = Minimum project total cost in (EGP), $P_{C\,\text{Max}}$ = Maximum project total cost in (EGP), $P_{Q\,\text{Min}}$ = Minimum project quality in (%), $P_{Q\,\text{Max}}$ = Maximum project quality in (%).

➢ Smart Critical Path Method System “SCPMS”

Smart Critical Path Method System ‘‘SCPMS’’ software is designed by C++ programming code system to provide a number of new and unique capabilities, including: (1) ranking the obtained optimal plans according to a set of planner specified weights that represent the relative importance of time, cost, and quality in the analysed project; (2) visualizing and viewing the generated optimal trade-off between construction duration, cost, and quality according to planner ranking relative weights to facilitate the selection of an optimal plan that considers the specific project needs; and (3) providing seamless integration with available project management calculations to benefit from their practical project scheduling and control features. In order to provide the aforementioned capabilities of SCPMS software, the system is implemented as shown in Fig. 6 and developed in four main modules, as shown in Fig. 7: (1) database module to facilitate the storage and retrieval of construction scheduling, resource utilization, and optimal trade-off data; (2) running module to provide a seamless integration of the project database with processing module and smart optimization model and responsible for all calculations runs; (3) processing module to do all calculations; and (4) userinterface module to facilitate the input of project data and the visualization of the three- dimensional time-cost-quality trade-off generated by the system.

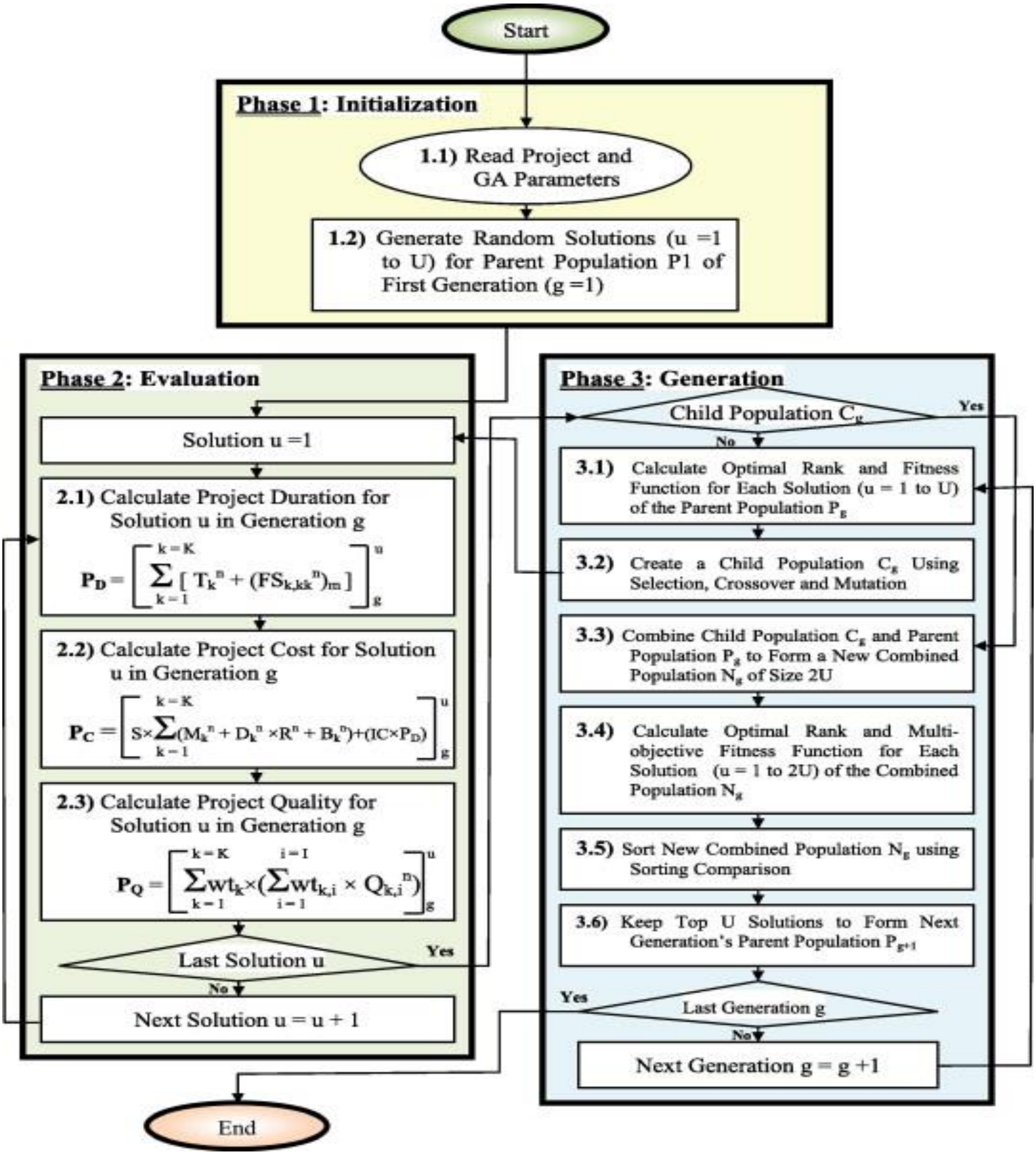

Start
Phase 1: Initialization
1.1) Read Project and GA Parameters
1.2) Generate Random Solutions (u =1 to U) for Parent Population P1 of First Generation (g =1)
Phase 2: Evaluation
Solution u =1
2.1) Calculate Project Duration for Solution u in Generation g
P_D = [Σ_{k=1}^{k=K} [T_k^n + (FS_{k,kk}^n)_m]]_g^u
2.2) Calculate Project Cost for Solution u in Generation g
P_C = [S×Σ_{k=1}^{k=K}(M_k^n + D_k^n ×R^n + B_k^n)+(IC×P_D)]_g^u
2.3) Calculate Project Quality for Solution u in Generation g
P_Q = [Σ_{k=1}^{k=K} wt_k×(Σ_{i=1}^{i=I} wt_{k,i} × Q_{k,i}^n)]_g^u
Last Solution u
Yes
No
Next Solution u = u + 1
Phase 3: Generation
Child Population C_g
Yes
No
3.1) Calculate Optimal Rank and Fitness Function for Each Solution (u = 1 to U) of the Parent Population P_g
3.2) Create a Child Population C_g Using Selection, Crossover and Mutation
3.3) Combine Child Population C_g and Parent Population P_g to Form a New Combined Population N_g of Size 2U
3.4) Calculate Optimal Rank and Multi-objective Fitness Function for Each Solution (u = 1 to 2U) of the Combined Population N_g
3.5) Sort New Combined Population N_g using Sorting Comparison
3.6) Keep Top U Solutions to Form Next Generation's Parent Population P_{g+1}
Yes
Last Generation g
No
Next Generation g = g +1
End

Figure 6 ‘‘SCPMS’’ model implementation.

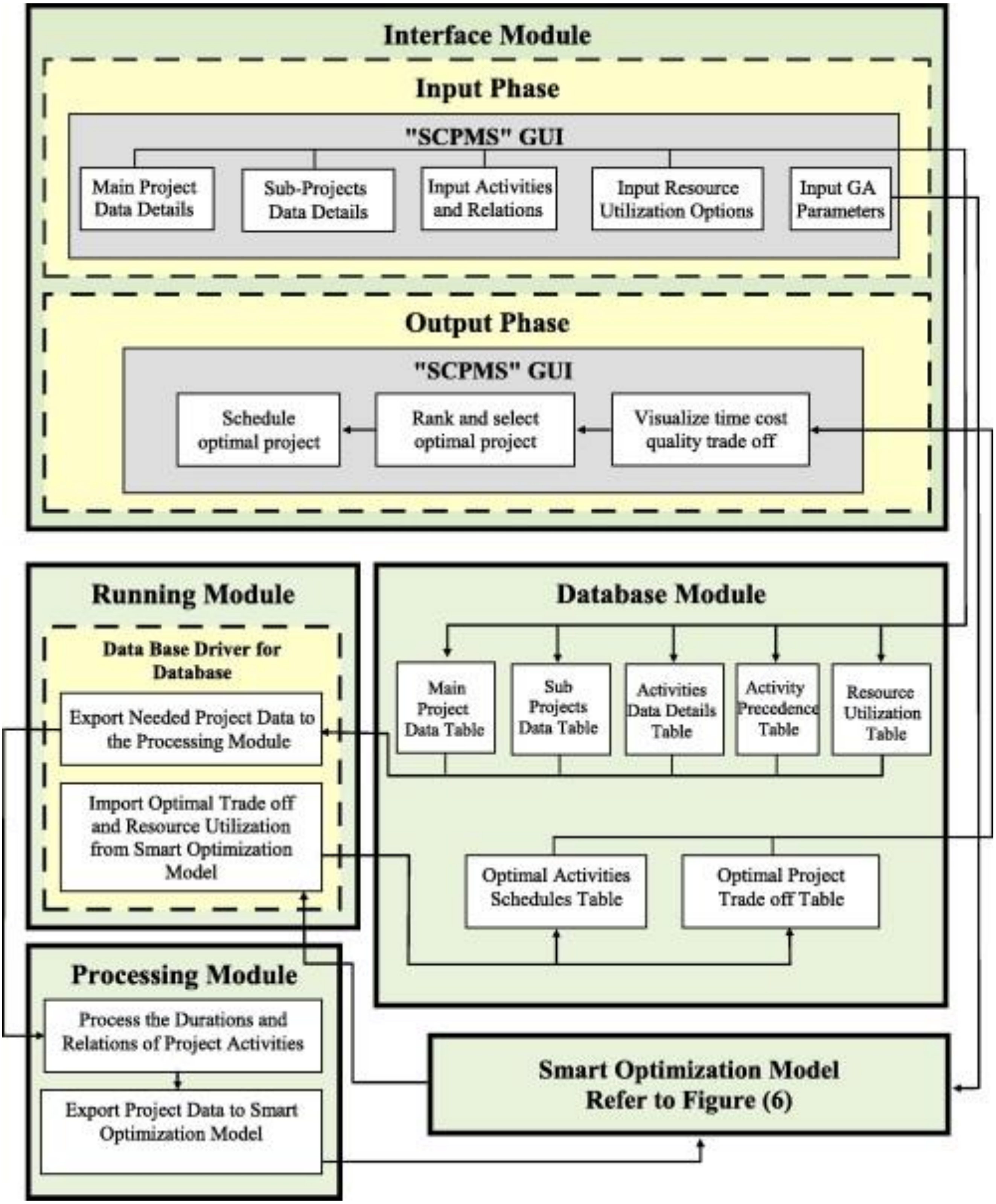

Interface Module
Input Phase
"SCPMS" GUI
Main Project Data Details
Sub-Projects Data Details
Input Activities and Relations
Input Resource Utilization Options
Input GA Parameters
Output Phase
"SCPMS" GUI
Schedule optimal project
Rank and select optimal project
Visualize time cost quality trade off
Running Module
Data Base Driver for Database
Export Needed Project Data to the Processing Module
Import Optimal Trade off and Resource Utilization from Smart Optimization Model
Database Module
Main Project Data Table
Sub Projects Data Table
Activities Data Details Table
Activity Precedence Table
Resource Utilization Table
Optimal Activities Schedules Table
Optimal Project Trade off Table
Processing Module
Process the Durations and Relations of Project Activities
Export Project Data to Smart Optimization Model
Smart Optimization Model
Refer to Figure (6)

Figure 7 Main modules of ‘‘SCPMS’’ system.

Database module:

The main purpose of this module is to develop a database that is capable of storing the necessary input data (e.g., project data details, project activities and available resource utilization options) and the produced output data (e.g., generated optimal trade-off between construction duration, cost, and quality). It was composed of nine main tables that were designed to store the following information: (1) project data details table; (2) project holidays data table; (3) project exception data table; (4) project activities table; (5) successors relationships between activities table; (6) available resource utilization options for each activity table; (7) quality importance weight for each activity table; (8) optimal activities schedules and optimal resource utilization option for each activity table; and (9) optimal project time, cost and quality trade-off. A schematic representation of these database tables and the relationships between them was shown using an entity dataflow diagram (see Fig. 8).

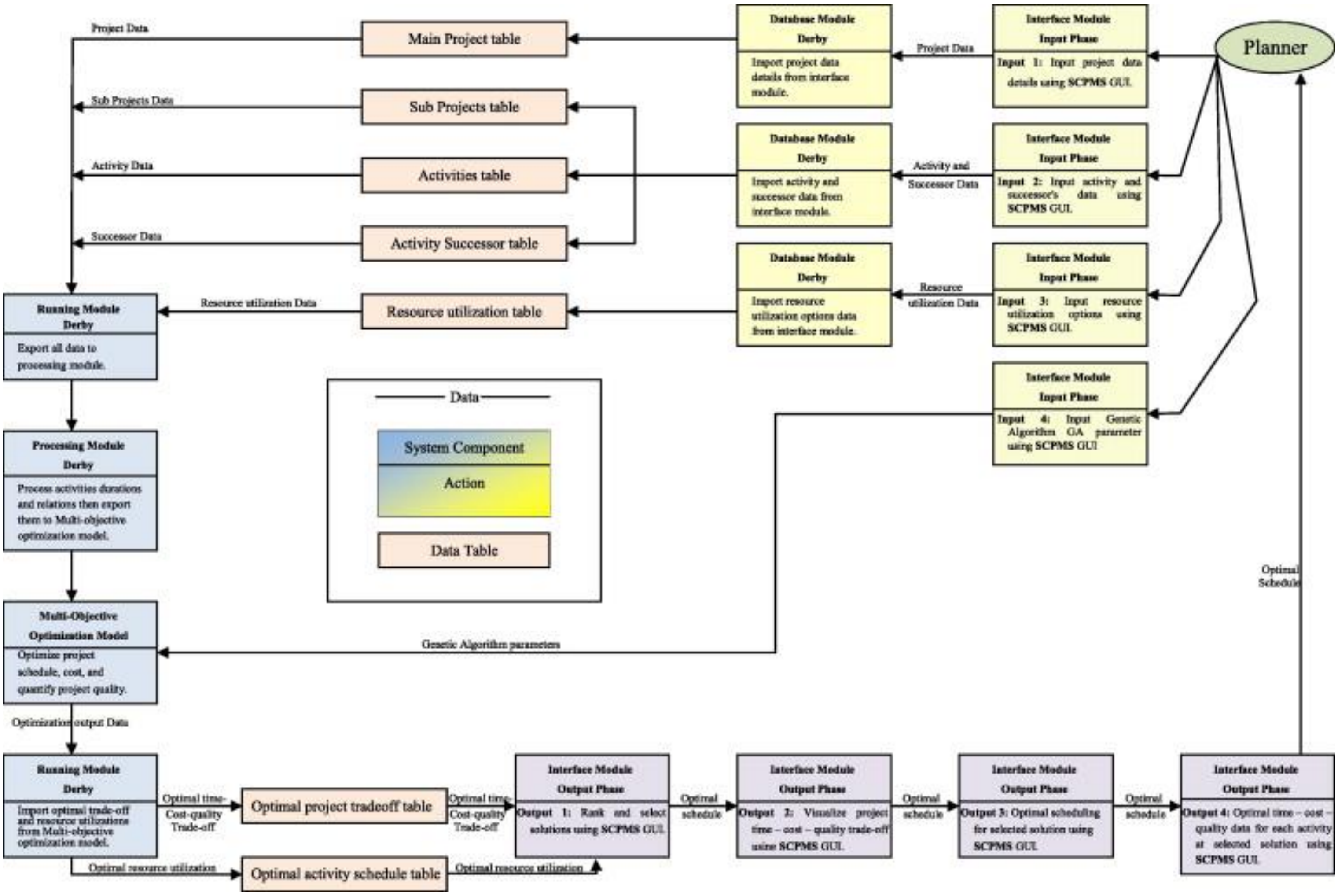

Project Data
Main Project table
Database Module
Derby
Import project data details from interface module.
Project Data
Interface Module
Input Phase
Input 1: Input project data details using SCPMS GUI
Planner
Sub Projects Data
Sub Projects table
Activity Data
Activities table
Database Module
Derby
Import activity and successor data from interface module.
Activity and Successor Data
Interface Module
Input Phase
Input 2: Input activity and successor's data using SCPMS GUI
Successor Data
Activity Successor table
Database Module
Derby
Import resource utilization options data from interface module.
Resource utilization Data
Interface Module
Input Phase
Input 3: Input resource utilization options using SCPMS GUI
Running Module
Derby
Export all data to processing module.
Resource utilization Data
Resource utilization table
Interface Module
Input Phase
Input 4: Input Genetic Algorithm GA parameter using SCPMS GUI
Data
System Component
Action
Data Table
Processing Module
Derby
Process activities durations and relations then export them to Multi-objective optimization model.
Multi-Objective
Optimization Model
Optimize project schedule, cost, and quantify project quality.
Genetic Algorithm parameters
Optimal Schedule
Optimization output Data
Running Module
Derby
Import optimal trade-off and resource utilizations from Multi-objective optimization model.
Optimal time-Cost-quality Trade-off
Optimal project tradeoff table
Optimal time-Cost-quality Trade-off
Interface Module
Output Phase
Output 1: Rank and select solutions using SCPMS GUI
Optimal schedule
Interface Module
Output Phase
Output 2: Visualize project time – cost – quality trade-off using SCPMS GUI
Optimal schedule
Interface Module
Output Phase
Output 3: Optimal scheduling for selected solution using SCPMS GUI
Optimal schedule
Interface Module
Output Phase
Output 4: Optimal time – cost – quality data for each activity at selected solution using SCPMS GUI
Optimal resource utilization
Optimal activity schedule table
Optimal resource utilization

Figure 8 “SCPMS’’ dataflow diagram

Running module:

Running can be classified as a programming code especially C++ coding system and its applications that were designed to allow, communicate and exchange different data from available modules. It was developed in ‘‘SCPMS’’ software to enable the integration of database module, user interface module, processing module and smart optimization model. To accomplish this, the running module was designed by C++ programming code as follows: (1) the C++ programming code driver; and (2) the Derby data base driver. First, the C++ programming code driver is utilized to perform two main steps: (1) export scheduling data from the project, activity, successors, and resource utilization tables in the database module to processing module; and (2) import the generated results from smart optimization model to the optimal activity schedules and optimal project trade-off tables in the database module. Second, the derby driver is used in ‘‘SCPMS’’ software to perform two main steps: (1) export the existing project scheduling data from the ‘‘SCPMS’’ software input interface module to the activities and successors relationship table in the database module; (2) import the selected optimal scheduled data obtained by smart optimization model from the database module to ‘‘SCPMS’’ software output interface module to facilitate its visualization and storage. The main data were transferred using the two drivers in running module in ‘‘SCPMS’’ software by user interface modules.

Processing module:

Processing module can be classified as a programming code especially C++ coding system and its applications, that are designed to run all calculations into studied project by integration of the running module and the smart optimization model. To accomplish this, the processing module is designed to utilize the C++ programming code driver as follows: (1) calculate the scheduled data exported by running module from the project, activity, successors, and resource utilization tables in the database module to processing module; and (2) import the scheduled data of project, activity, successors, and resource utilization tables to smart optimization model. The main data transferred using the C++ programming code driver in processing module are scheduled data in a database module in ‘‘SCPMS’’ software by user interface module.

User interface module:

The user interface module was implemented by ‘‘SCPMS’’ software to facilitate the input construction planning data and the output of generated optimal schedules. The module was designed to implement necessary interface functions in two main stages: (1) an input stage that facilitates the input of project data details, project activities, activities relations, activities importance weights, available resource utilization options, and genetic algorithm parameters; and (2) an output stage that allows the user to rank and visualize the optimal project scheduling solutions and optimal activity resource utilization options obtained by smart optimization model. In each of previous two stages, the module was designed to enable a graphical interface with construction planners by using developed Graphical User Interface ‘‘GUI’’ forms in ‘‘SCPMS’’ software. The GUI forms are used to facilitate: (1) the initial project schedule creation; and (2) the generated optimal schedule presentation, analysis, and control. On the other hand, developed GUI forms were implemented in ‘‘SCPMS’’ software by using C++ programming code in order to facilitate the integration of other modules in ‘‘SCPMS’’ software. The interactions between the input and output stages in proposed module and others remaining modules in ‘‘SCPMS’’ software were illustrated using a standard dataflow diagram, as shown in Fig. 8. The following two sections describe the data flow between the construction planner and the different components of ‘‘SCPMS’’ software during the input and output stages, which are executed after the start of program as declared in ‘‘SCPMS’’ software welcome form.

Input stage:

Input stage Input stage was designed to facilitate input of data in ‘‘SCPMS’’ software. To accomplish this, a group of interactive GUI forms were developed to guide the planner in entering data by four main steps that were designed to: (1) allow adding, deleting and editing mega construction projects by group of buttons; (2) allow editing or inserting the data details of studied project, which includes ID, name, location, etc.; (3) allow adding, deleting and editing project activities; (4) allow editing or inserting the data details of the project activity, which includes ID, description, quantity, etc.; (5) allow editing or inserting activity resource details and utilization options; (6) allow adding, deleting and editing the relationships between project activities; (7) allow editing or inserting the data details of two sequential activities relationship; (8) allow adding, deleting and editing of project holidays; (9) allow editing or inserting the project holidays data details; (10) allow adding, deleting and editing of project exceptions by using group of buttons; (11) allow editing or inserting the data details of the project exceptions; and (12) allow entering genetic algorithm parameters. Final step of input stage at proposed user interface module was designed to facilitate the input of genetic algorithm parameters that are needed to initiate smart optimization model. Main parameters of genetic algorithm include: (1) population size; (2) generations number; (3) crossover type (simple, double, multiple); (4) crossover probability (0.00: 1.00); (5) mutation probability (0.00: 1.00); and (6) random number used to create first population of solutions. The smart optimization process is then invoked by clicking the optimization button in the project details form. After optimization process, the obtained results are then visualized and ranked at output stage of proposed module.

output stage:

Output stage of proposed user interface module was designed to facilitate viewing and visualization of the generated optimal project time, cost, and quality trade-off. To accomplish this, output stage was executed in five main steps that were designed to: (1) rank the generated optimal solutions by using a group of weights which specify the relative importance of time, cost, and quality to decision makers, and allow scrolling through ranked optimal solutions; (2) display obtained project time cost-quality trade-off and allow scrolling them through generated solutions; (3) scheduling the selected main project solution and sub-projects; (4) scheduling related activities by plotting bar chart view; and (5) printing data details of optimized selected mega construction project solution.

➢ case study

A project case study is analysed in order to illustrate the use of the present smart optimization model and demonstrate its capabilities. The case study consists of 18 construction activities as shown in Fig. 9; each one has a number of possible resource utilization options that can be used to construct the activity as shown in Table 3. The case study was originally introduced by Feng et al. [29], to illustrate construction time cost trade-off analysis, and therefore it did not include data on construction quality. In order to highlight the capabilities of proposed model, the data of the original case study are expanded to illustrate the impact of each resource utilization option on construction quality in addition to its time and cost as shown in Table 3. The estimated activity performance in time, cost, and quality indicators for each resource utilization option can be obtained by performing a statistical analysis of its historical performance in previous projects, assuming that these performance levels can be used to predict their future performance. This simple case study had an average of 3.4 resource utilization options to construct each of the 18 activities as shown in Fig. 9 and Table 3, which produces about 3 billion (i.e., 3.418) possible combinations for delivering the entire project. Each of these possible combinations leads to a unique impact on project performance toward time, cost, and quality, and the main challenge here is to search this large search space to find solutions that establish an optimal and delicate balance between construction time, cost and quality. The proposed optimization model was used to search this large space of possible solutions using the developed smart optimization model.

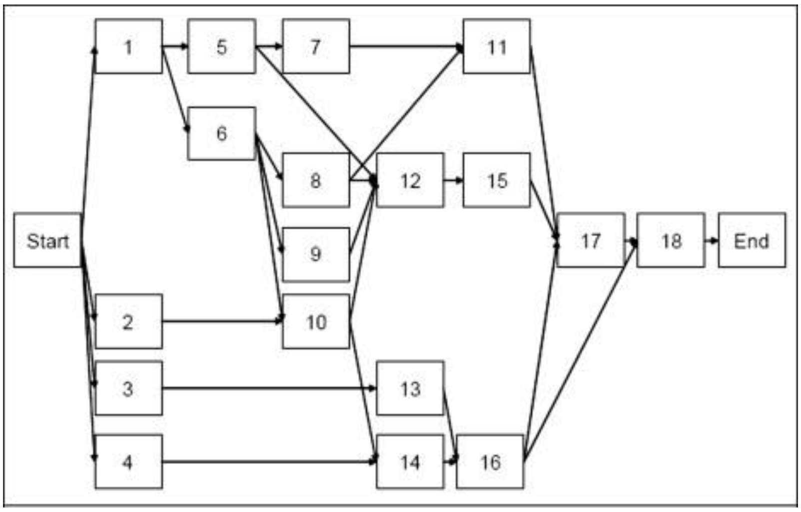


Figure 9 Case study network.

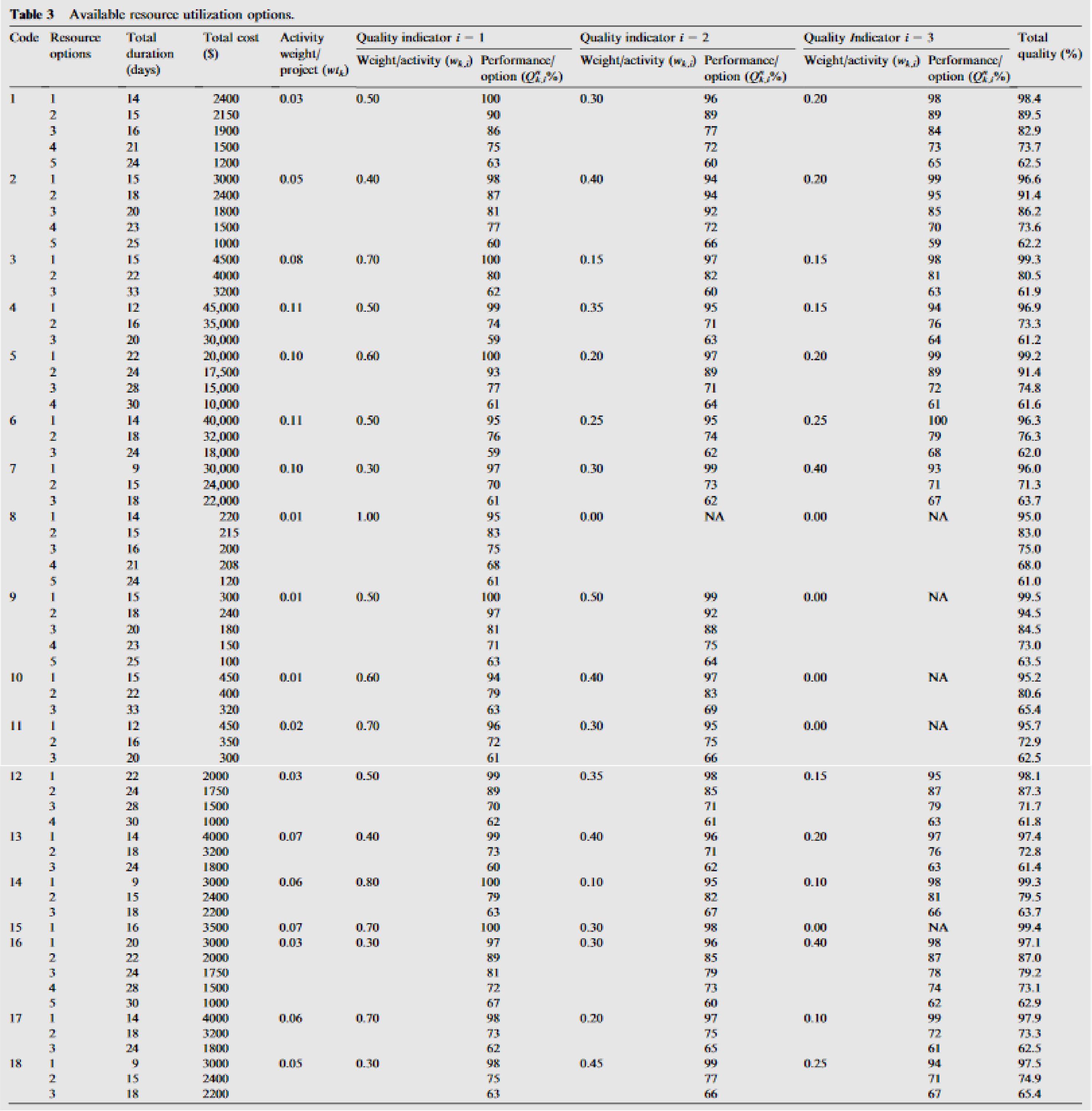


Table 3 Available resource utilization options.

Code	Resource options	Total duration (days)	Total cost ($)	Activity weight/ project (wt_k)	Quality indicator $i = 1$		Quality indicator $i = 2$		Quality Indicator $i = 3$		Total quality (%)
					Weight/activity ($w_{k,i}$)	Performance/ option ($Q^{n}_{k,i}$/%)	Weight/activity ($w_{k,i}$)	Performance/ option ($Q^{n}_{k,i}$/%)	Weight/activity ($w_{k,i}$)	Performance/ option ($Q^{n}_{k,i}$/%)	
1	1	14	2400	0.03	0.50	100	0.30	96	0.20	98	98.4
	2	15	2150			90		89		89	89.5
	3	16	1900			86		77		84	82.9
	4	21	1500			75		72		73	73.7
	5	24	1200			63		60		65	62.5
2	1	15	3000	0.05	0.40	98	0.40	94	0.20	99	96.6
	2	18	2400			87		94		95	91.4
	3	20	1800			81		92		85	86.2
	4	23	1500			77		72		70	73.6
	5	25	1000			60		66		59	62.2
3	1	15	4500	0.08	0.70	100	0.15	97	0.15	98	99.3
	2	22	4000			80		82		81	80.5
	3	33	3200			62		60		63	61.9
4	1	12	45,000	0.11	0.50	99	0.35	95	0.15	94	96.9
	2	16	35,000			74		71		76	73.3
	3	20	30,000			59		63		64	61.2
5	1	22	20,000	0.10	0.60	100	0.20	97	0.20	99	99.2
	2	24	17,500			93		89		89	91.4
	3	28	15,000			77		71		72	74.8
	4	30	10,000			61		64		61	61.6
6	1	14	40,000	0.11	0.50	95	0.25	95	0.25	100	96.3
	2	18	32,000			76		74		79	76.3
	3	24	18,000			59		62		68	62.0
7	1	9	30,000	0.10	0.30	97	0.30	99	0.40	93	96.0
	2	15	24,000			70		73		71	71.3
	3	18	22,000			61		62		67	63.7
8	1	14	220	0.01	1.00	95	0.00	NA	0.00	NA	95.0
	2	15	215			83					83.0
	3	16	200			75					75.0
	4	21	208			68					68.0
	5	24	120			61					61.0
9	1	15	300	0.01	0.50	100	0.50	99	0.00	NA	99.5
	2	18	240			97		92			94.5
	3	20	180			81		88			84.5
	4	23	150			71		75			73.0
	5	25	100			63		64			63.5
10	1	15	450	0.01	0.60	94	0.40	97	0.00	NA	95.2
	2	22	400			79		83			80.6
	3	33	320			63		69			65.4
11	1	12	450	0.02	0.70	96	0.30	95	0.00	NA	95.7
	2	16	350			72		75			72.9
	3	20	300			61		66			62.5
12	1	22	2000	0.03	0.50	99	0.35	98	0.15	95	98.1
	2	24	1750			89		85		87	87.3
	3	28	1500			70		71		79	71.7
	4	30	1000			62		61		63	61.8
13	1	14	4000	0.07	0.40	99	0.40	96	0.20	97	97.4
	2	18	3200			73		71		76	72.8
	3	24	1800			60		62		63	61.4
14	1	9	3000	0.06	0.80	100	0.10	95	0.10	98	99.3
	2	15	2400			79		82		81	79.5
	3	18	2200			63		67		66	63.7
15	1	16	3500	0.07	0.70	100	0.30	98	0.00	NA	99.4
16	1	20	3000	0.03	0.30	97	0.30	96	0.40	98	97.1
	2	22	2000			89		85		87	87.0
	3	24	1750			81		79		78	79.2
	4	28	1500			72		73		74	73.1
	5	30	1000			67		60		62	62.9
17	1	14	4000	0.06	0.70	98	0.20	97	0.10	99	97.9
	2	18	3200			73		75		72	73.3
	3	24	1800			62		65		61	62.5
18	1	9	3000	0.05	0.30	98	0.45	99	0.25	94	97.5
	2	15	2400			75		77		71	74.9
	3	18	2200			63		66		67	65.4

➢Added value

The model was able to significantly reduce this large space by precluding dominated solutions in the successive generations of the evolutionary algorithm, using the Pareto optimality principles shown in Table 3. This led to the selection of 305 Pareto optimal solutions for this application case study. Each of these solutions identifies an optimal resource utilization option for each of the 18 construction activities, and accordingly it provides a unique and optimal trade-off between project time, cost and quality. A sample of these solutions and their impact on project performance toward time, cost, and quality is shown in Table 4.

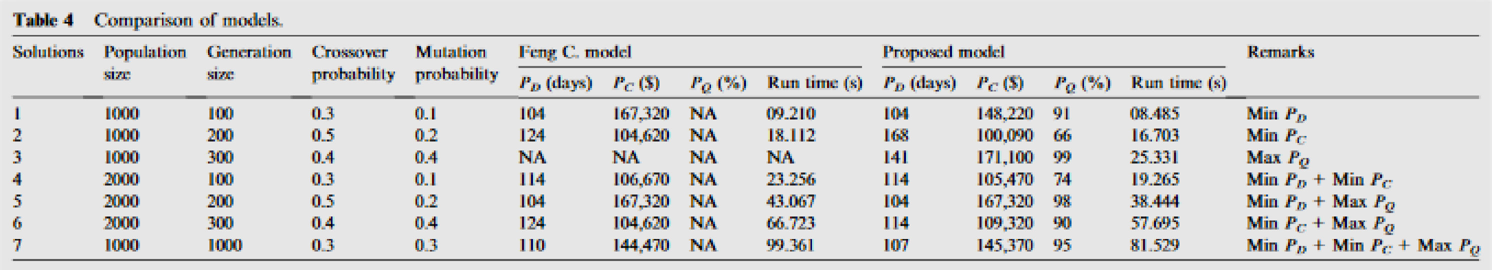


Table 4 Comparison of models.

Solutions	Population size	Generation size	Crossover probability	Mutation probability	Feng C. model				Proposed model				Remarks
					P_D (days)	P_C ($)	P_Q (%)	Run time (s)	P_D (days)	P_C ($)	P_Q (%)	Run time (s)	
1	1000	100	0.3	0.1	104	167,320	NA	09.210	104	148,220	91	08.485	Min P_D
2	1000	200	0.5	0.2	124	104,620	NA	18.112	168	100,090	66	16.703	Min P_C
3	1000	300	0.4	0.4	NA	NA	NA	NA	141	171,100	99	25.331	Max P_Q
4	2000	100	0.3	0.1	114	106,670	NA	23.256	114	105,470	74	19.265	Min P_D + Min P_C
5	2000	200	0.5	0.2	104	167,320	NA	43.067	104	167,320	98	38.444	Min P_D + Max P_Q
6	2000	300	0.4	0.4	124	104,620	NA	66.723	114	109,320	90	57.695	Min P_C + Max P_Q
7	1000	1000	0.3	0.3	110	144,470	NA	99.361	107	145,370	95	81.529	Min P_D + Min P_C + Max P_Q

The comparison showed that the efficiency of proposed model was higher than Feng et al. Furthermore, the comparison illustrated the effectiveness of taking short run time using the same processor, compared to the run time of the Feng et al. model that has a fluctuating efficiency. This demonstrates that the present model is capable of robustly solving the time-cost trade-off problem, while providing the added capability of considering and quantifying quality and visualizing tradeoff between project time, cost and quality for the analyzed case study.

➢ Possibility of applying this optimization model in Morocco

According to The BNC Network, Morocco accounts over than 278 mega construction projects, valued at approximately USD 114 Billion. In many sectors as the urban construction projects, the industrial projects, oil and gas projects, transports projects and utilities projects.

In my opinion I think that applying this optimization model to the local mega construction projects is quite possible, and especially advantageous for the investors by realizing the optimal choices that minimize construction time and cost while maximizing its quality.

➢ Conclusion

This model was designed to search for the optimal resource utilization plans that minimize construction time and cost while maximizing its quality, and also to make development in three main tasks. First task, the model is formulated to incorporate all major decision variables and to enable the optimization of the three major objectives in this resource utilization problem. Second task, the model is formulated to enable quantifying construction quality in the optimization process, and then the model is implemented as an advanced multi-objective evolutionary algorithm in third task. Smart Critical Path Method System, ‘‘SCPMS’’ was developed in four main tasks that led to the development of: (1) database module to store the project data details, project activities details, activities relations, activities resource utilization options and genetic algorithm parameters; (2) running module to provide calculations with a seamless integration of database module with user interface module, processing module and the smart optimization model; (3) processing module was designed to perform all calculations and was developed to enable the integration of the database module with smart optimization model; and (4) user interface module to facilitate the input of project data and genetic algorithm parameters, as well as the visualization and ranking of the obtained optimal solutions. An application case study was analyzed to illustrate the use of proposed model and demonstrate its capabilities in considering quality in the optimization process and in developing optimal tradeoff between construction time, cost and quality. These new proposed models should prove usefulness to decision makers, especially those who are involved in new types of contracts requiring high quality performance in mega construction projects.

➢References

1- Complexity of Construction Mega Infrastructure Project https://www.hindawi.com/journals/complexity/2018/7382083/

2- Optimizing performance in infrastructure project delivery | McKinsey https://www.mckinsey.com/business-functions/operations/our-insights/optimizing-performance-in-infrastructure-project-delivery#

3- Reasons why megaprojects fail | UCL News - UCL – University College London https://www.ucl.ac.uk/news/2020/feb/reasons-why-megaprojects-fail

4- State housing and infrastructure projects in Morocco drive growth in construction and real estate | Morocco 2015 | Oxford Business Group https://oxfordbusinessgroup.com/overview/state-housing-and-infrastructure-projects-morocco-drive-growth-construction-and-real-estate

5- On going industrial projects in Morocco https://www.bncnetwork.net/sector-country/Industrial/Morocco

6- optimization for mega construction https://www.researchgate.net/publication/26312718

7- Mega Projects Optimization | WashingtonRise https://washingtonrise.com/insight/mega-projects-optimization/